REMARKS

Status of the application

The pending claims stand rejected in view of prior art. The claims were previously amended to further distinguish Applicant's invention. The claims have been amended herein to place the claims in better form for consideration on a possible appeal. Applicant requests entry of the amendments pursuant to 37 C.F.R 1.116(b)(2). In addition, Applicant now offers additional remarks and comments, for clarifying the claimed invention. Reexamination and reconsideration are respectfully requested.

General

1. Specification

The Office Action objects to the specification for failing to provide proper antecedent basis for the term "machine-implemented method". Claims 42 and 60 have been amended to recite "computer-implemented method" instead of "machine-implemented method". Applicant respectfully requests entry of these amendments pursuant to 37 C.F.R 1.116(b)(2).

2. Claim objection

Claim 42 is objected to because it does contain the connector "and" before the last claim limitation. Claim 42 has been amended to include the connector. Applicant respectfully requests entry of this amendment pursuant to 37 C.F.R 1.116(b)(2).

Prior art rejections

1. First Section 103 rejection

The pending claims stand rejected under 35 U.S.C. § 103(a) as being unpatentable over S. Chakravarthy et al "Composite Events for Active Databases: Semantics, Contexts and Detection", Proc. of the 20th VLDB Conf, Santiago, Chile, 1994, pp. 606-617, pp. 42-48 (hereinafter "Chakravarthy") in view of Etzion et al U.S. Patent No. 6,604,093 (hereinafter "Etzion") and Hellerstein et al U.S. Patent Pub. No. 2002/0165842 (hereinafter "Hellerstein"). This rejection is respectfully traversed.

The Office Action likens Applicant's claimed invention to Chakravarthy's composite event detector, but acknowledges that Chakravarthy and Etzion do not include any teaching of the following highlighted portions of Applicant's claim 42:

> 42. A computer-implemented method for managing event-condition-action rules in a database system, the method comprising the computer-implemented steps performed by said database system of:
>
> **storing, in a database managed by said database system,** rule data that defines a composite event comprised of two or more primitive events, at least one condition related to the composite event, and at least one action to be performed upon satisfaction of said at least one condition;
>
> detecting a first database event as an occurrence of a first one of the primitive events;
>
> determining whether the first database event satisfies a first sub-condition of said at least one condition, wherein said rule data indicates that satisfaction of said first sub-condition is not sufficient to satisfy said at least one condition;
>
> **persistently storing in the database results data that indicates that said first sub-condition was satisfied by said first database event;**
>
> detecting a second database event as an occurrence of a second one of the primitive events;
>
> **reading said results data from said database;**
>
> determining whether the at least one condition is satisfied based on the results **data read from the database** and the second database event.

Nonetheless, the Office Action adds Hellerstein for the prospect that it teaches those highlighted claim limitations, through Hellerstein's system for processing historical event data.

At the outset, it is important to understand that Applicant does not claim the notion of Event-Condition-Action processing. Nor does Applicant claim the notion of storing event data in a database. Instead, what Applicant claims is a novel technique for incremental evaluation of conditions with respect to primitive events that comprise a composite event. Specifically, Applicant's claim 42 features (1) durably storing results data indicating that a first sub-condition of a condition related to a composite event is satisfied by a first database event, and then (2) determining whether the condition as a whole is satisfied based on a second detected database

event and the durably stored results data. In contrast to conventional composite event detection mechanisms, the approach of Applicant's claims is not constrained by the amount of physical memory available to store a set of Event-Condition-Action rules, thereby facilitating processing of much larger sets of rules.

This is not the same as conventional composite event detection mechanisms, such as described in Chakravarthy and Etzion. In conventional composite event detection, there is a focus on speed in processing events against a rule set. These conventional mechanisms store data structures for processing events against rule sets in physical memory (e.g., volatile Random Access Memory (RAM) or main memory of a computer) where they can be accessed much more quickly than if stored on a larger but slower storage medium such as a hard disk. For example, Chakravarthy describes data structures such as an "event graph" and "operator trees" that are implemented in physical memory of a computer (see, e.g., Chakravarthy, p. 615 stating "The local composite event detector and the application share the same address space and our event detector uses an event graph similar to operator trees.")) In Etzion, in-physical-memory linked list data structure are used to detect composite event occurrences (see, e.g., Etzion, FIG. 3 illustrating a data structure into which the Etzion system maps event instances it receives). To support larger ECA rule sets than can be supported by mechanisms limited by the amount of physical memory, Claim 1 involves durably storing the results of such incremental evaluations in a database. The Office Action acknowledges that Chakravarthy and Etzion do not teach Applicant's technique for incremental evaluation, so the point need not be belabored.

However, the Office Action seems to believe that Applicant's technique can be recreated simply by bolting on Hellerstein's system for processing historical event data onto some combination of Chakravarthy's and Etzion's systems for composite event detection. However, Hellerstein's system does not provide enough teaching to convert a Chakravarhty-Etzion system into one that supports Applicant's technique for incremental evaluation of conditions with respect to primitive events that comprise a composite event. Importantly, Applicant's claim limitations are in terms of persistently storing results data, not event data.

For example, consider the following language of Applicant's claim 42:

persistently storing in the database **results data** that indicates that said first sub-condition was satisfied by said first database event;

detecting a second database event as an occurrence of a second one of the primitive events;

reading said **results data** from said database; and

determining whether the at least one condition is satisfied based on the **results data** read from the database and the second database event.

(Emphasis added.)

As shown by the foregoing claim limitations, Applicant's invention of claim 42 is directed to (1) durably storing results data indicating that a first sub-condition of a condition related to a composite event is satisfied by a first database event, and then (2) determining whether the condition as a whole is satisfied based on a second detected database event and the durably stored results data.

That Hellerstein does not teach or suggest such a concept is made clear by the Hellerstein reference itself. For example, Hellerstein (Paragraph [0041]) states:

> In step 302, the event management decision support system reads previously accumulated **event data** into an event cache. The previously accumulated data is stored in memory associated with the event management decision support system, e.g., Event DB 180 in FIG. 1, prior to being read into the event cache. The previously accumulated event data represents **historical event data**. It is to be understood that the term "historical," as used herein, refers to **event data that was generated by network devices and received by the event management system at some prior time**. The time period from which the data is drawn may depend on the event management application. Thus, for example, the event data may be data generated and received between a point in time in the immediate past and some earlier relative point in time. Therefore, the historical event data accumulated over the desired time period is read from the Event DB into the event cache of the event management decision support system. It is this event data that is used to generate the one or more correlation rules.
>
> (Emphasis added.)

The event data stored in the Event DB described in Paragraph [0041] refers to data that was generated by network devices. The event data does not refer to results data that indicates that a sub-condition of a condition related to a composite event was satisfied. In other words, the event data represents an occurrence of an event, not whether an event satisfied a sub-condition of a condition related to a composite event. Consequently, the combination of Chakravarthy, Etzion, and Hellerstein asserted in the Office Action does not satisfy at least the following feature of claim 42 when taken as a whole:

persistently storing in the database results data that indicates that said first sub-condition was satisfied by said first database event;

Moreover, Hellerstein describes an expert system for offline construction of correlation rules for event management; it does not describe techniques for runtime detection of composite event occurrences within a database system. The focus of Hellerstein is on the construction of correlation rules by analyzing historical event data. It involves a human analyzing historical event data or a computer executing data mining algorithms on historical event data stored in an event cache.

Applicant's claims, in contrast, focuses on runtime detection of composite event occurrences within a database system. Significantly, Applicant's claims provide a design that supports the processing of detected events against large rule sets – larger than those that can be supported by conventional composite event detection systems constrained by the amount of available physical memory. While Hellerstein's system may be helpful in construction of those rule sets, it does not provide any teaching about how to scale the processing of events against rule sets. Consequently, Applicant's approach of claim 42 for runtime evaluation of a rule set takes up where Hellerstein's approach for creating a rule set leaves off.

Applicant's other independent claim recites limitations similar to those recited in claim 42. Therefore, the remarks above with respect to claim 42 apply equally to claim 51.

For the reasons discussed, it is respectfully submitted that the claims set forth a patentable advance over the art. In view of the clarifying remarks above, it is believed that the claims distinguish over the combined references, and that any rejection under Section 103 is overcome.

2. Remaining claims

The pending claims not discussed so far are dependant claims that depend on an independent claim that is discussed above. Because each dependant claim includes the features of claims upon which they depend, the dependant claims are patentable for at least those reasons the claims upon which the dependant claims depend are patentable. Removal of the rejections with respect to the dependant claims and allowance of the dependant claims is respectfully requested. In addition, the dependent claims introduce additional features that independently render them patentable. Due to the fundamental differences already identified, a separate discussion of those features is not included at this time.

Conclusions

For the reasons set forth above, all of the pending claims are now in condition for allowance. The Examiner is respectfully requested to contact the undersigned by telephone relating to any issue that would advance examination of the present application.

A petition for extension of time, to the extent necessary to make this reply timely filed, is hereby made. If applicable, a check for the petition for extension of time fee and other applicable fees is enclosed herewith. If any applicable fee is missing or insufficient, throughout the pendency of this application, the Commissioner is hereby authorized to charge any applicable fees and to credit any overpayments to our Deposit Account No. 50-1302.

Respectfully submitted,

HICKMAN PALERMO TRUONG & BECKER LLP

Dated: June 30, 2009

/AdamCStone#60531/

Adam Stone
Reg. No. 60,531

2055 Gateway Place Suite 550
San Jose, California 95110-1093
Telephone No.: (408) 414-1080
Facsimile No.: (408) 414-1076